



INSTITUTO FEDERAL

Catarinense

Campus Camboriú

Listas

Professora: Lissandra Maiara Fischer

lissandra.fischer@ifc.edu.br

Objetivo:



- Compreender o conceito de Listas.

O que vimos até agora

```
print("cáculo do fatorial")
n = int(input("Insira um número inteiro não negativo"))
fat = 1
for i in range(1, n+1):
    fat = fat * i
    print(fat)
print(n, "! =", fat)
```

Listas e Vetores

- Em **Python** existem três tipos principais de variáveis compostas: **Listas, Tuplas e Dicionários**, além de *arrays*.

Vamos aos conceitos:

- **Listas** são um tipo de variável que permite armazenar vários valores, acessados por um índice. Estes valores podem ser de um mesmo tipo ou de **tipos diversos**.
- **Vetores** são um tipo de variável que permite armazenar vários valores, acessados por um índice. Estes valores **devem ser do mesmo tipo**.

Ou seja, listas são mais flexíveis que vetores.

Como imaginar uma lista:

- Imagine um **edifício** que contém apartamentos, no térreo é o andar 0, o primeiro andar é o andar 1 e assim por diante;
- O índice é utilizado para especificarmos o “**apartamento**” onde guardamos os nossos dados;
- Em um prédio de **6 andares** teremos nosso índice variando de **0 até 5**;

- Se chamarmos nosso prédio de **P**, teremos **P[0], P[1], P[2], P[3], P[4], P[5]**;

Neste caso **P** é o nome da nossa lista e o **número entre colchetes** é o índice;

P

Outra forma de representar um vetor ou lista?

L

0	1	2	3	4	5	6	7	8	9

Listas são mais **flexíveis** que prédios e podem crescer e ou diminuir com o tempo.

Como declarar uma lista?

```
L = []
```

- Este comando cria uma lista denominada de **L** e os colchetes (`[]`) após o símbolo de igualdade indicam que é uma lista vazia;

```
Z = [15, 8, 9]
```

- Este comando cria uma lista denominada de **Z** e esta lista contém 3 elementos: **15, 8 e 9**

Como acessar um elemento de uma lista?

```
Z = [15, 8, 9]
```

- Se acessarmos `Z[0]` o resultado que será exibido será 15, pois estamos acessando o primeiro elemento das listas, ou seja, o elemento que está armazenado no índice 0.
- `Z[1]` será igual a 8 e `Z[2]` será igual a 9.

Como alterar um elemento de uma lista?

```
Z = [15, 8, 9]
```

- Se atribuirmos `Z[0] = 10` o conteúdo de `Z[0]` será atualizado. E ficaremos com:

```
Z = [10, 8, 9]
```

Como ler e mostrar uma lista?

```
# cálculo das médias com notas digitadas
```

```
notas = [0, 0, 0, 0, 0]
```

```
soma = 0
```

```
x = 0
```

```
while x < 5:
```

```
    notas[x] = float(input(f"Nota {x}:"))
```

```
    soma += notas[x]
```

```
    x = x + 1
```

```
x = 0
```

```
while x < 5:
```

```
    print(f"Nota {x}: {notas[x]:6.2f}")
```

```
    x += 1
```

```
print(f"Média: {soma/ x:5.2f}")
```

é preciso iniciar com zeros(0) neste caso

precisamos de estrutura de repetição para percorrer cada posição da lista.

atribuindo valores float para a lista

exibindo o conteúdo de cada posição da lista;

Como verificar o tamanho da lista?

```
Z = [15, 8, 9]
```

- Podemos utilizar a função **len()**;

```
print(len(Z))
```

- Será exibido o valor 3, o que corresponde ao número de elementos da lista;

Adição de elementos em uma lista

```
Z = [15, 8, 9]
```

- Para adicionarmos um elemento ao final da lista podemos utilizar o método **append**;

```
Z.append(23)
```

- Outras formas de adicionar elementos a lista:

```
Z += [2] ou Z = Z + [2]
```

Removendo elementos de uma lista

```
Z = [15, 8, 9]
```

- Para removermos um elemento da lista podemos utilizar a instrução **del**;

```
del Z[1]
```

- irá ficar:

```
Z = [15, 9]
```

Importante observar que os elementos removidos da lista não ocuparão mais espaço e os índices serão reorganizados.

Podemos apagar também fatias da lista:

```
del Z[1:99] restaria Z = [0, 99, 100]
```

Referências



Referências Básicas

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. **Lógica de programação: a construção de algoritmos e estruturas de dados**. 3. ed. Pearson Prentice Hall. 2005

MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de.. **Algoritmos: lógica para desenvolvimento de programação de computadores..** 27. ed.. Érica. 2014

Referências Complementares

DOWNEY, Allen B. **Pense em Python**. 2ª Ed. Novatec. 2016

MENEZES, Nilo Ney de Coutinho. **Introdução a programação com Python**. 3ª Ed. Novatec. 2019

CORMEN, Thomas H et al. **Algoritmos: teoria e prática**. 2. ed. Elsevier, Campus,. 2002

Referências na Internet

<https://docs.python.org/3/>

<https://www.w3schools.com/python/default.asp>